ESD-TDR-65-561

Semiannual Technical Summary

Graphics

30 November 1965

Prepared for the Advanced Research Projects Agency
under Electronic Systems Division Contract AF 19(628)-5167 by

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Lexington, Massachusetts

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology; this work was supported by the U.S. Advanced Research Projects Agency of the Department of Defense under Air Force Contract AF 19(628)-5167 (ARPA Order 691).

28

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

GRAPHICS

SEMIANNUAL TECHNICAL SUMMARY REPORT
TO THE
ADVANCED RESEARCH PROJECTS AGENCY

1 JUNE – 30 NOVEMBER 1965

ISSUED 17 DECEMBER 1965

LEXINGTON MASSACHUSETTS

ABSTRACT

Recent efforts in the Graphics Program have concentrated on the development of a Graphical Service System, display routines compatible with the new time-sharing system, APEX, and a universal translator, VITAL, which will be used to generate a graphical compiler. Programs for clipping and approximating conic segments have been developed and initial experiments which apply graphical techniques to procedure description have been attempted.

Accepted for the Air Force
Stanley J. Wisniewski
Lt Colonel, USAF
Chief, Lincoln Laboratory Office

INTRODUCTION

The Graphics Program is a direct outgrowth of continuous research in man-machine communications which has been active since the inception of TX-2. This report describes the first six months' activity under ARPA sponsorship.

Present efforts are concentrated on the development of a Graphical Service System,[1] display routines compatible with the new time-sharing system, APEX,[2] and a universal translator, VITAL, which will be used to generate a graphical compiler.

In addition, programs providing for the clipping and approximating of conic segments have been developed using a generalized parametric matrix representation.[3]

Initial experiments applying graphical techniques to procedure description, rather than data input, have proved encouraging and further work is contemplated.

A preliminary study has been undertaken to explore the possibility of networking three time-sharing systems (Lincoln, SDC, and MAC) in order to experiment with sharing graphics software.

Currently, three consoles are available on TX-2 consisting of scope, keyboard, and typewriter. All scopes are driven from a single generator which can draw lines and parabolas,[4] and one of them has a charactron for fast text display. The present vector generator is now overloaded at current drawing speeds. An additional generator is being considered, as well as modifications in design to increase generator operating speed.

Jack I. Raffel

TABLE OF CONTENTS

GRAPHICS

I. APEX DISPLAY PACKAGE

We have, at the moment, two types of display users: one who wishes merely to see an alpha-numeric graphical display and the other who wants to feed back information "through" the display, using an interrupt device such as a light pen. The present display package adequately serves the needs of the former, although modification of the character set for speed and aesthetic appeal continues. The bulk of the present effort, however, is directed toward the latter group.

Display regeneration is handled by keeping all active display files permanently in core and "round robin" file switching is now in use. A portion of the display file itself is used for stacking light pen hits, knob positions, real time, tracking data, and control parameters so that switching files via a one-register change in the SPAT LBM[5] memory sets up the program for the next user.

The present system includes knob and pen activated user interrupt for inactive users (shade mode). A facility for interrupt of an active program (via Ghost Map) will be implemented soon. The pen-tracking program will be improved and will be able to cause an interrupt when tracking status changes from "normal" to "fast", "slow", and/or "lost", as the user directs. The user also defines the pen velocities for "fast" and "slow".

A display tracking program has been developed which predicts a velocity for the pen as well as the average radius of the moving elliptical aperture of the pen. With this predicted radius, less time and fewer display points are used to find the outer edges of the aperture, which, in turn, are used to determine the new center of the aperture. The velocity is determined by means of a second-order prediction. By using a fixed time interval, the last two vectors are saved and the new position is determined by adding three times the last vector to the starting position of the second-last vector.

Two pen stacks are used per file, and they are automatically exchanged when the user requests pen data. The file is always available to the user in read-only form and can be changed only through the APEX display calls. When a new file is presented to the system, it is checked for changes and only unsullied files are accepted. Users are expected to build and change files only through use of the display calls.

II. GRAPHICAL SERVICE SYSTEM

The COLLECTOR routine is the part of the Graphical Service System which serves as a control interface for the man at the display console. The input/output routines of APEX will be used by the COLLECTOR which (for the present) will be treated by APEX as a private user.

The input information which will be processed by the COLLECTOR will be obtained by the following anticipated equipment:

Light pen

- Existing display data seen by the pen
- Location and motion of the pen

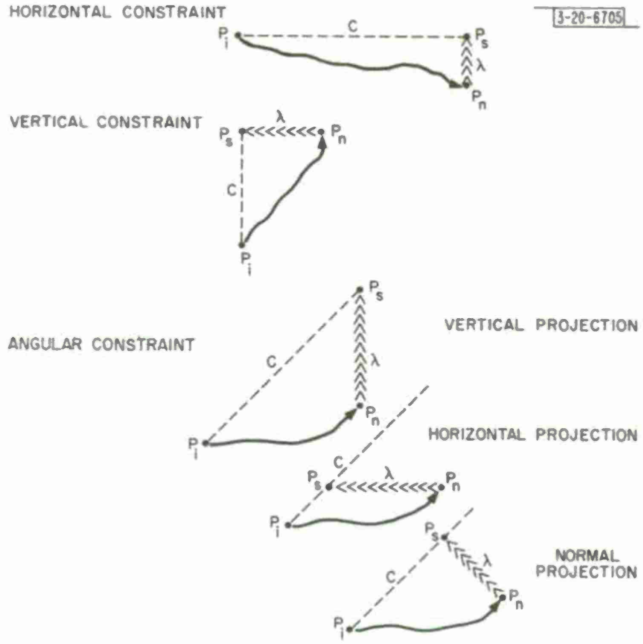

Fig. 1. Constraints for pseudopointer.

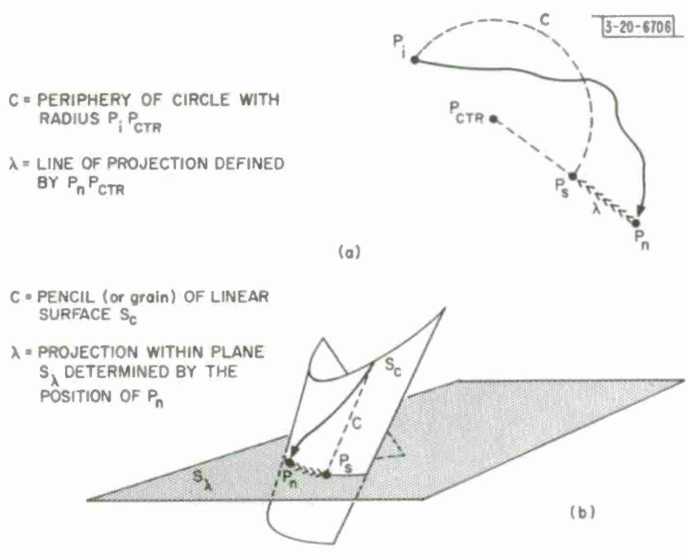

Fig. 2. Examples of constraints: (a) circular, (b) surface.

Analog-to-digital knobs

Lincoln Writer keyboard

Special push-button panel (including foot pedals and pen switches)

RAND Tablet pointer

Existing display data as "seen via a comparator" by the pointer
Location and motion of the pointer

3D pointer

Existing display data as "seen via a comparator" by the pointer
Location and motion of the pointer

Remote computer/display console via data telephone line.

It is planned that the graphics package will allow individual graphic control languages to be used. However, basic control algorithms will be provided by the collector in a modular form so that individual users may take advantage of them as part of their language. For the most part, these algorithms have to do with feedback control for providing convenient operation at the scope console.

Pseudopointer:– One of the more interesting sets of algorithms will control the motion of a pseudopoint based on the motion of whatever physical pointer is being used as a driver. Let

P_i = initial position of both pointer and pseudopointer

P_n = new position of pointer

P_s = new position of pseudopointer

C = constraint for motion of P_s

λ = projection of P_n.

A series of constraint algorithms will be available, including those shown in Figs. 1 and 2(a-b).

Motion of Pen as Input Signal:– In an attempt to eliminate awkward button-pushing procedures and to deal with the problem of sharing the time of a central processor, a plan has been devised whereby the COLLECTOR will use the motion of the pen as input control information. The APEX display executive will alert the COLLECTOR if the physical pointer has changed speeds; the speeds are slow, normal, and fast. During "normal" motion of the pointer, the COLLECTOR will, in effect, ignore the action. During "slow" motion of the pointer, the COLLECTOR will monitor the pointer at some arbitrary real-time interval and, if any constraining operators are in effect, will calculate and display the position of the pseudopoint. In addition, the COLLECTOR will make a test of how stationary the pointer is and, based on some arbitrary criteria for "stop", will execute whatever "stop" actions are specified by the individual graphics control language. By displaying the pseudopoint only during slow motion of the pointer and allowing display changes only when the pointer has stopped, the load on the central processor is reduced considerably. The signal that the pointer is moving fast may be used for several purposes. One which we anticipate is in removing automatic pointer constraints. For example, if a user has automatic constraints in effect, any time he moves from some initial pointer position P_i to a new position P_n which lies within some relative band or area of constraint (Fig. 3), the constraining operation will be executed (during slow speed).

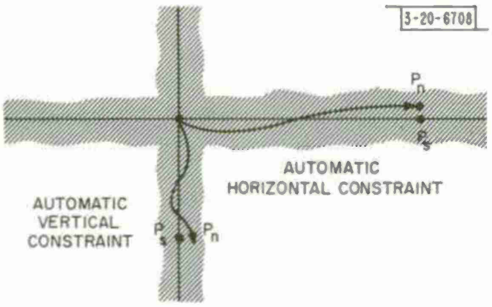

Fig. 3. Example of automatic horizontal and vertical constraints.

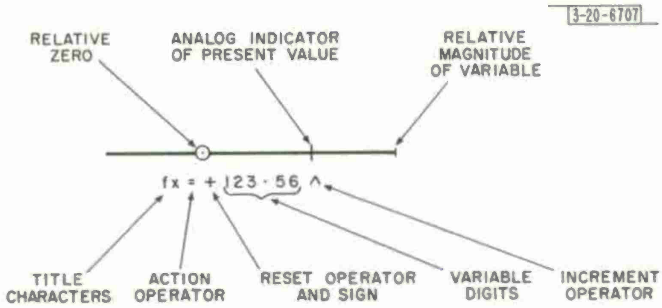

Fig. 4. Representation of variables.

Another example of automatic constraint is the use of the coarse grid. For example, if a switch is being drawn and the user desires to keep his lengths in intervals of only $\frac{1}{16}$ inch, anytime the pseudopointer is displayed it is constrained to the nearest $\frac{1}{16}$-inch grid intersection from the pointer's position. During a drawing operation, the user may find a situation in which he does not want the automatic constraint. Rather than requiring an override button, one could cause the pen to wiggle momentarily, thus giving a "fast" motion signal to the COLLECTOR to shut off the automatic constraint until a new P_i is initiated.

Control of Variable Parameters and Light Button:– The use of light buttons or "targets" will be controlled by the COLLECTOR and it is anticipated that mnemonic characters rather than points (which require plastic overlays) will be used. Also, the COLLECTOR would deal with "pages" of related buttons, thereby allowing considerable flexibility to the designers of individual graphic control languages. The present display consoles of the TX-2 allow data to be "vector-walked" off the normal drawing area of the CRT, thereby allowing light buttons and the digital values of dynamic parameters to be displayed without interfering with the actual graphic presentation.

A variable (Fig. 4) will consist of title characters, an action operator, a reset operator and sign, variable digits (with a radix point if required), and an analog indicator. For example, if any one of the title characters is touched by the pointer, the variable will be displayed, if off, and turned off, if on. At the moment there are three action operators:

≡ means change the variable as the digits are changed

= means the digits may be changed without changing the actual variable – at the same time, the = will be changed to a → when the digits are touched

→ means the variable is not equal to current value displayed by the digits.

If ≡ is pointed at, it becomes =

If = is pointed at, it becomes ≡

If → is pointed at, it becomes = and the variable is set to the value displayed by the digits

If the reset operator which is + or – is pointed at, the digits will be changed to some preset value

If the increment operator ∧ is pointed at, it will be changed to ∨, the decrement operator, and vice versa.

If the digits are pointed at, they will be incremented or decremented with appropriate carries when necessary. Maximum and minimum limits may be specified and, when exceeded, digits will be reset to the limit value. As the variable is changed, the analog indicator will show its relative value with respect to its full range. When the display consoles are modified, the variable will be able to be changed by pushing and pulling this indicator. Control or parameter values will then be achieved by a combination of knobs, digital pointer action, and "throttle" pointer action.

The COLLECTOR has had one version operating and is currently being written for full operating capability. The problem of interfacing with the graphic control language interpreter is now being studied.

III. VITAL

The TX-2 compiler-compiler, VITAL, is nearing completion. All the basic parts of the system are written and largely checked out. Current efforts are directed toward various auxiliary features such as editing and checkout packages.

The entire system is designed to operate in a multiuser environment. A VITAL user will have a great variety of programming capabilities at his disposal. Besides the usual advantages of interactive checkout of programs, the user will be able to apply these same mechanisms to changing the language in which he is programming. Extensive use will be made of the TX-2 display capability to improve reaction times.

Documentation at various levels is also in preparation. We expect that VITAL will be a fully functional system in the next reporting period.

IV. CONIC DISPLAY PROGRAMS

A. CLIPPER

CLIPPER examines an object represented by a jxn matrix A ($j = 1$, 2, or 3 for point, line, or conic, respectively) and determines what parts of the object lie within a volume represented by a matrix Vo. This "clipping" may be done in spaces that have arbitrary finite dimensions and produces a clipped object composed of convex cells with planar boundaries. All mathematical computations are performed in a homogeneous coordinate system.

B. PARABOLIZER

PARABOLIZER is a subroutine for generating conic sections through simulation, using parabolic arcs. By making the starting speed of an arc equal to the ending speed of the previous arc, and supplying the rates of change in the speeds and the time for each arc, a continuous smooth curve can be generated. This procedure makes the most efficient use of scope hardware, because the user supplies the starting X, Y position and starting X, Y speeds for the first arc only, and then supplies just the rate of change in the speeds and the time for each succeeding section of curve.

Each conic to be simulated is represented as a 3×3 homogeneous parametric matrix. The homogeneous coordinate technique simplifies the representation and processing of graphical information by reducing operations to standard matrix forms and techniques. For instance, a

conic represented in matrix form may be rotated or translated simply by multiplying by a single 3×3 transformation matrix. The homogeneous coordinate technique can be thought of as the addition of an extra coordinate to each vector, a scale factor, so that the vector has the same meaning after multiplication by a constant.

Simulation of conic sections by using parabolic arcs combined with the homogeneous coordinate technique has many distinct advantages over the normal straight-line simulation approach. First of all, it is not necessary to return to the implicit form of the conic, determine the type of conic, and then, using a special set of equations for the particular type of conic, generate a display list of many straight lines. Neither is it necessary to solve a set of very complex differential or simultaneous equations. The 3×3 homogeneous parametric matrix and a single set of simple derived equations allows the direct determination of a display list of parabolas. Second, the curve is displayed as a continuous curve, using parabolas with much less data supplied to the scope. The curve is completely continuous because the first derivative at the end of one parabolic arc is the same as the starting first derivative of the next arc, with only a new second derivative and time being supplied at the beginning of each succeeding arc. Complete circles have been generated which require only eight parabolic arcs, but as many as fifty straight lines may be required. An increase in the number of parabolic segments does not appear to improve the shape of the curve and appears only to increase both processing and display generation times. A circle of eight parabolas requires only an eighteen-word display list, whereas the fifty straight lines would require over one hundred data words. Thus, parabolization not only gives a better looking curve, but also allows faster generation.

In general, the conic is parameterized over the time interval 0 to 1. The ideal situation is to have the conic generated as fast as possible by adjusting the time and speed so that the speed is as close to maximum (maximum = 1) as possible. On any section of the curve where the speed falls below a predetermined threshold, the speeds are rescaled and the times adjusted to meet these criteria. The overall effect of this rescaling is that the curve is drawn as fast as possible with constant intensity.

V. GRAPHICALLY DESCRIBED PROCEDURES

During the past year, an experimental system for programming graphically has been developed for the TX-2 computer. A two-dimensional program description may be created on the computer display scope by interactive Sketchpad-like controls.[6] This graphically described program may then be executed. During execution, each active element is made to flash when in use, thus making it possible to follow the course of computation activity. A variety of debugging features has been developed for the kind of graphical programs allowed within the experimental system.

The results obtained from the experimentation with graphical programming may be summarized as follows:

> Standard notions of program flow may be substantially modified. The kind of graphical programs developed are controlled by data flow through the program. Explicit flow control is generally not needed, but may be included at the user's option.
>
> Since the program execution is controlled by data connections, flow, when present, will be treated as another type variable in the same way, for example, as variables are classified as integer or boolean.

With or without explicit flow control, a graphical program is a natural way of specifying parallel processes. The two-dimensionality of the program format is convenient for representing simultaneous actions.

Graphical programs lend themselves naturally to a number of debugging techniques. Some of these features should be applicable to flow-chart displays of conventional programs.

Detailed results of this research are being presented to M.I.T. as a doctoral thesis by W. R. Sutherland entitled "The On-line Graphical Specification of Computer Procedures."

VI. HARVARD SYNTACTIC ANALYSIS PROGRAM

A project has been initiated to investigate the application of graphical techniques to syntactic analysis. The first phase, which consists of re-programming the Harvard Syntactic Analysis Program, the dictionary and grammar lookup, and update, is almost completed. The syntactic phase and the display will be implemented in the near future. The bulk of this work is being done by consultants.

VII. COMPUTER NETWORK

An initial study has been undertaken by Computer Corporation of America for Lincoln Laboratory to determine the feasibility of linking three independent time-sharing systems (SDC, MAC, and Lincoln). The aim of this work is to study the problems and potential gains to be realized by providing dynamic access to programs located at remote centers. In particular, this project will emphasize mutual interchange of graphics software, an area in which related work is under way on all three systems.

VIII. DISPLAY HARDWARE

An analog generator capable of forming rotated cubics, parabolas and/or vectors for real-time display on a CRT has been built and placed on-line. The functions are generated by successive integration. Analog switching is done with field effect transistors at microsecond speeds. Although the analog generator is fast enough for driving a single CRT, multiple console operation requires a faster generator, now under study. It is also planned to include dividers in the system so affine transformations can take place. This allows the formation of all conics and the generation of perspective drawings.

An analog comparator has also been built and successfully used in conjunction with the existing analog display generator and interval timer. The comparator is used to determine when an indicator print becomes coincident with part of a displayed curve. This equipment is necessary to provide pointing capability for the RAND Tablet.

REFERENCES

1. L.G. Roberts, "A Graphical Service System with Variable Syntax," to be published in the 1965 Proceedings of ACM Programming Language Conference.
2. J.W. Forgie, "A Time- and Memory-Sharing Executive Program for Quick-Response, On-Line Applications," to be published in the Proceedings of the 1965 Fall Joint Computer Conference.
3. L.G. Roberts, "Homogeneous Matrix Representation and Manipulation of N-Dimensional Constructs," unpublished course notes.
4. T.E. Johnson, "Analog Generator for Real-Time Display of Curves," Technical Report 398, Lincoln Laboratory, M.I.T. (28 July 1965), DDC 623945.
5. Division 2 Quarterly Progress Report, Lincoln Laboratory, M.I.T. (15 February 1965), p. 1, DDC 612541.
6. I.E. Sutherland, "Sketchpad: A Man-Machine Graphical Communication System," Technical Report 296, Lincoln Laboratory, M.I.T. (30 January 1963), DDC 404459.

DOCUMENT CONTROL DATA – R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Lincoln Laboratory, M.I.T.	2a. REPORT SECURITY CLASSIFICATION Unclassified	
	2b. GROUP None	
3. REPORT TITLE Semiannual Technical Summary Report to the Advanced Research Projects Agency on Graphics		
4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)* Semiannual Technical Summary Report 1 June to 30 November 1965		
5. AUTHOR(S) *(Last name, first name, initial)* Raffel, Jack I.		
6. REPORT DATE 30 November 1965	7a. TOTAL NO. OF PAGES 16	7b. NO. OF REFS 6
8a. CONTRACT OR GRANT NO. AF 19(628)-5167	9a. ORIGINATOR'S REPORT NUMBER(S) Semiannual Technical Summary (30 November 1965)	
b. PROJECT NO. ARPA Order 691		
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* ESD-TDR-65-561	
d.		
10. AVAILABILITY/LIMITATION NOTICES None		
11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency, Department of Defense	

13. ABSTRACT

Recent efforts in the Graphics Program have concentrated on the development of a Graphical Service System, display routines compatible with the new time-sharing system, APEX, and a universal translation, VITAL, which will be used to generate a graphical compiler. Programs for clipping and approximating conic segments have been developed and initial experiments which apply graphical techniques to procedure description have been attempted.

14. KEY WORDS

Graphical communication
TX-2
time-sharing
man-machine
display systems
Sketchpad

Printed by
United States Air Force
L. G. Hanscom Field
Bedford, Massachusetts